

Intel

```

88888 000 88888 5555555 A
8 8 0 0 8 8 5 A A
8 8 0 0 0 8 5 A A
88888 0 0 0 88888 555555 AAAAAA
8 8 0 0 0 8 8 5 A A
8 8 0 0 8 8 5 A A
88888 000 88888 555555 A A

```

8085A MICROPROCESSOR Instruction Set Summary

--> X1	1	40	Vcc (+5V)
--> X2	2	39	HOLD <--
<-- RESET OUT	3	38	HLDA -->
<-- SOD	4	37	CLK (OUT) -->
--> SID	5	36	RESET IN <--
--> TRAP	6	35	READY <--
--> RST 7.5	7	34	IO/ \overline{M} -->
--> RST 6.5	8	33	S1 -->
--> RST 5.5	9	32	\overline{RD} -->
--> INTR	10	31	\overline{WR} -->
<-- \overline{INTA}	11	30	ALE -->
<--> AD0	12	29	S0 -->
<--> AD1	13	28	A15 -->
<--> AD2	14	27	A14 -->
<--> AD3	15	26	A13 -->
<--> AD4	16	25	A12 -->
<--> AD5	17	24	A11 -->
<--> AD6	18	23	A10 -->
<--> AD7	19	22	A9 -->
Vss	20	21	A8 -->

8085A

Written by Jonathan Bowen
 Programming Research Group
 Oxford University Computing Laboratory
 8-11 Keble Road
 Oxford OX1 3QD
 England

Tel +44-865-273840

Created May 1983
 Updated April 1985
 Issue 1.1

Copyright (C) J.P.Bowen 1985

Mnemonic	Op	SZAPC	~s	Description	Notes
ACI n	CE	*****	7	Add with Carry Immediate	A=A+n+CY
ADC r	8F	*****	4	Add with Carry	A=A+r+CY(21X)
ADC M	8E	*****	7	Add with Carry to Memory	A=A+[HL]+CY
ADD r	87	*****	4	Add	A=A+r (20X)
ADD M	86	*****	7	Add to Memory	A=A+[HL]
ADI n	C6	*****	7	Add Immediate	A=A+n
ANA r	A7	****0	4	AND Accumulator	A=A&r (24X)
ANA M	A6	****0	7	AND Accumulator and Memory	A=A&[HL]
ANI n	E6	**0*0	7	AND Immediate	A=A&n
CALL a	CD	-----	18	Call unconditional	-[SP]=PC, PC=a
CC a	DC	-----	9	Call on Carry	If CY=1(18~s)
CM a	FC	-----	9	Call on Minus	If S=1 (18~s)
CMA	2F	-----	4	Complement Accumulator	A=~A
CMC	3F	-----*	4	Complement Carry	CY=~CY
CMP r	BF	*****	4	Compare	A-r (27X)
CMP M	BF	*****	7	Compare with Memory	A-[HL]
CNC a	D4	-----	9	Call on No Carry	If CY=0(18~s)
CNZ a	C4	-----	9	Call on No Zero	If Z=0 (18~s)
CP a	F4	-----	9	Call on Plus	If S=0 (18~s)
CPE a	EC	-----	9	Call on Parity Even	If P=1 (18~s)
CPI n	FE	*****	7	Compare Immediate	A-n
CPO a	E4	-----	9	Call on Parity Odd	If P=0 (18~s)
CZ a	CC	-----	9	Call on Zero	If Z=1 (18~s)
DAA	27	*****	4	Decimal Adjust Accumulator	A=BCD format
DAD B	09	-----*	10	Double Add BC to HL	HL=HL+BC
DAD D	19	-----*	10	Double Add DE to HL	HL=HL+DE
DAD H	29	-----*	10	Double Add HL to HL	HL=HL+HL
DAD SP	39	-----*	10	Double Add SP to HL	HL=HL+SP
DCR r	3D	****-	4	Decrement	r=r-1 (0X5)
DCR M	35	****-	10	Decrement Memory	[HL]=[HL]-1
DCX B	0B	-----	6	Decrement BC	BC=BC-1
DCX D	1B	-----	6	Decrement DE	DE=DE-1
DCX H	2B	-----	6	Decrement HL	HL=HL-1
DCX SP	3B	-----	6	Decrement Stack Pointer	SP=SP-1
DI	F3	-----	4	Disable Interrupts	
EI	FB	-----	4	Enable Interrupts	
HLT	76	-----	5	Halt	
IN p	DB	-----	10	Input	A=[p]
INR r	3C	****-	4	Increment	r=r+1 (0X4)
INR M	3C	****-	10	Increment Memory	[HL]=[HL]+1
INX B	03	-----	6	Increment BC	BC=BC+1
INX D	13	-----	6	Increment DE	DE=DE+1
INX H	23	-----	6	Increment HL	HL=HL+1

8085A.TXT

INX SP	33	-----	6	Increment Stack Pointer	SP=SP+1
JMP a	C3	-----	7	Jump unconditional	PC=a
JC a	DA	-----	7	Jump on Carry	If CY=1(10~s)
JM a	FA	-----	7	Jump on Minus	If S=1 (10~s)
JNC a	D2	-----	7	Jump on No Carry	If CY=0(10~s)
JNZ a	C2	-----	7	Jump on No Zero	If Z=0 (10~s)
JP a	F2	-----	7	Jump on Plus	If S=0 (10~s)
JPE a	EA	-----	7	Jump on Parity Even	If P=1 (10~s)
JPO a	E2	-----	7	Jump on Parity Odd	If P=0 (10~s)
JZ a	CA	-----	7	Jump on Zero	If Z=1 (10~s)
LDA a	3A	-----	13	Load Accumulator direct	A=[a]
LDAX B	0A	-----	7	Load Accumulator indirect	A=[BC]
LDAX D	1A	-----	7	Load Accumulator indirect	A=[DE]
LHLD a	2A	-----	16	Load HL Direct	HL=[a]
LXI B,nn	01	-----	10	Load Immediate BC	BC=nn
LXI D,nn	11	-----	10	Load Immediate DE	DE=nn
LXI H,nn	21	-----	10	Load Immediate HL	HL=nn
LXI SP,nn	31	-----	10	Load Immediate Stack Ptr	SP=nn
MOV r1,r2	7F	-----	4	Move register to register	r1=r2 (1XX)
MOV M,r	77	-----	7	Move register to Memory	[HL]=r (16X)
MOV r,M	7E	-----	7	Move Memory to register	r=[HL] (1X6)
MVI r,n	3E	-----	7	Move Immediate	r=n (0X6)
MVI M,n	36	-----	10	Move Immediate to Memory	[HL]=n
NOP	00	-----	4	No Operation	
ORA r	B7	**0*0	4	Inclusive OR Accumulator	A=Avr (26X)
ORA M	B6	**0*0	7	Inclusive OR Accumulator	A=Av[HL]
ORI n	F6	**0*0	7	Inclusive OR Immediate	A=Avn
OUT p	D3	-----	10	Output	[p]=A
PCHL	E9	-----	6	Jump HL indirect	PC=[HL]
POP B	C1	-----	10	Pop BC	BC=[SP]+
POP D	D1	-----	10	Pop DE	DE=[SP]+
POP H	E1	-----	10	Pop HL	HL=[SP]+
POP PSW	F1	-----	10	Pop Processor Status word	{PSW,A}=[SP]+

Mnemonic	Op	SZAPC	~s	Description	Notes
PUSH B	C5	-----	12	Push BC	-[SP]=BC
PUSH D	D5	-----	12	Push DE	-[SP]=DE
PUSH H	E5	-----	12	Push HL	-[SP]=HL
PUSH PSW	F5	-----	12	Push Processor Status word	-[SP]={PSW,A}
RAL	17	-----*	4	Rotate Accumulator Left	A={CY,A}<-
RAR	1F	-----*	4	Rotate Accumulator Right	A=->{CY,A}
RET	C9	-----	10	Return	PC=[SP]+
RC	D8	-----	6	Return on Carry	If CY=1(12~s)
RIM	20	-----	4	Read Interrupt Mask	A=mask
RM	F8	-----	6	Return on Minus	If S=1 (12~s)
RNC	D0	-----	6	Return on No Carry	If CY=0(12~s)
RNZ	C0	-----	6	Return on No Zero	If Z=0 (12~s)
RP	F0	-----	6	Return on Plus	If S=0 (12~s)
RPE	E8	-----	6	Return on Parity Even	If P=1 (12~s)
RPO	E0	-----	6	Return on Parity Odd	If P=0 (12~s)
RZ	C8	-----	6	Return on Zero	If Z=1 (12~s)
RLC	07	-----*	4	Rotate Left Circular	A=A<-
RRC	0F	-----*	4	Rotate Right Circular	A=->A
RST z	C7	-----	12	Restart (3X7)	-[SP]=PC, PC=z
SBB r	9F	*****	4	Subtract with Borrow	A=A-r-CY
SBB M	9E	*****	7	Subtract with Borrow	A=A-[HL]-CY
SBI n	DE	*****	7	Subtract with Borrow Immed	A=A-n-CY
SHLD a	22	-----	16	Store HL Direct	[a]=HL
SIM	30	-----	4	Set Interrupt Mask	mask=A
SPHL	F9	-----	6	Move HL to SP	SP=HL
STA a	32	-----	13	Store Accumulator	[a]=A

8085A.TXT

STAX B	02	-----	7	Store Accumulator indirect	[BC]=A
STAX D	12	-----	7	Store Accumulator indirect	[DE]=A
STC	37	----1	4	Set Carry	CY=1
SUB r	97	*****	4	Subtract	A=A-r (22X)
SUB M	96	*****	7	Subtract Memory	A=A-[HL]
SUI n	D6	*****	7	Subtract Immediate	A=A-n
XCHG	EB	-----	4	Exchange HL with DE	HL<->DE
XRA r	AF	**0*0	4	Exclusive OR Accumulator	A=Axr (25X)
XRA M	AE	**0*0	7	Exclusive OR Accumulator	A=Ax[HL]
XRI n	EE	**0*0	7	Exclusive OR Immediate	A=Axn
XTHL	E3	-----	16	Exchange stack Top with HL	[SP]<->HL
PSW	-*01			Flag unaffected/affected/reset/set	
S	S			Sign (Bit 7)	
Z	Z			Zero (Bit 6)	
AC	A			Auxiliary Carry (Bit 4)	
P	P			Parity (Bit 2)	
CY	C			Carry (Bit 0)	
a p				Direct addressing	
M z				Register indirect addressing	
n nn				Immediate addressing	
r				Register addressing	
DB n(n)				Define Byte(s)	
DB 'string'				Define Byte ASCII character string	
DS nn				Define Storage Block	
DW nn(,nn)				Define Word(s)	
A B C D E H L				Registers (8-bit)	
BC DE HL				Register pairs (16-bit)	
PC				Program Counter register (16-bit)	
PSW				Processor Status word (8-bit)	
SP				Stack Pointer register (16-bit)	
a nn				16-bit address/data (0 to 65535)	
n p				8-bit data/port (0 to 255)	
r				Register (X=B,C,D,E,H,L,M,A)	
z				Vector (X=0H,8H,10H,18H,20H,28H,30H,38H)	
+	-			Arithmetic addition/subtraction	
&	~			Logical AND/NOT	
v	x			Logical inclusive/exclusive OR	
<-	->			Rotate left/right	
<->				Exchange	
[]				Indirect addressing	
[]+	-[]			Indirect address auto-inc/decrement	
{ }				Combination operands	
(X)				Octal op code where X is a 3-bit code	
If (~s)				Number of cycles if condition true	